Feature-Based Radial Distortion Correction for Image Mosaicing

Naoki Chiba
Mechatronics R. C.
SANYO Electric Co., Ltd.
chiba@mech.rd.sanyo.co.jp

Michihiko Minoh
Center for Information and Multimedia Studies
Kyoto University

Hiroshi Kano
Mechatronics R. C.
SANYO Electric Co., Ltd.

Abstract

We propose a feature-based method of correcting radial distortion for image mosaicing. Unlike conventional methods, our method does not require any calibration pattern or non-linear minimization. It can estimate the distortion parameter from images of natural scenes. Compared with previous methods not requiring a calibration pattern, our method computes the radial distortion parameter faster because it is based on feature correspondence. After establishing the feature correspondences by using optical flow estimation, we estimate the radial distortion coefficient in the process of estimating the homography between the images.

We show the effectiveness of our method by experiments both on real images and synthetic images. The accuracy of our method is comparable to the previous methods using a calibration pattern. The processing time is about four times faster than the previous methods which do not require a calibration pattern.

1. Introduction

Image mosaicing has become an active research area, because its ability to construct a large, high-resolution panoramic image from a collection of standard images. Applications include the construction of aerial and satellite photographs, photo editing, and the creation of virtual environments. Previous methods can be categorized into three types: direct methods, feature-based methods, and others.

Direct methods have been actively explored to estimate parametric transformations between images, [1], to create 2D and 3D aligned video mosaics, [16, 7, 9, 10], and to create full-view mosaics, [17].

Feature-based methods have been introduced to manage very large differences between images, [21], and to reduce the computational cost of estimating homographies between images with super resolution, [3].

The other type includes a real-time method using a simple transformation, [18], a method considering the forward movements of the camera, [19], and a phase-based method for the presence of moving objects, [5]

Radial distortion must be corrected for the lenses used on recent digital still cameras. The low cost of the lenses leads to significant distortion. In particular, zooming lenses and wide-angle lenses tend to contain severe distortion. This paper proposes a method of automatically correcting radial distortion for image mosaicing.

1.1. Related work

Conventional techniques for estimating lens distortion parameters can be categorized into three types. The first type uses special calibration patterns. Tsai [14] proposed a method using calibration patterns for providing 3D coordinates to estimate not only radial distortion parameters but also camera parameters such as focal length, image center and aspect ratio. This method requires careful operation to provide exact 3D locations. A recent method proposed in [20] is more flexible, because the images can be taken with a hand-held camera. This method, however, still requires a planar calibration pattern.

The second type requires straight lines in the image, although it does not require calibration patterns. Brown [2] uses a number of parallel plumb lines to compute radial distortion parameters using an iterative gradient-descent technique. Since the extraction of points on the plumb lines is manual-intensive, Swaminathan and Nayar proposed a method with a user-guided self-calibration approach in [11]. This method uses points picked by the user, along projections of straight lines in the image.

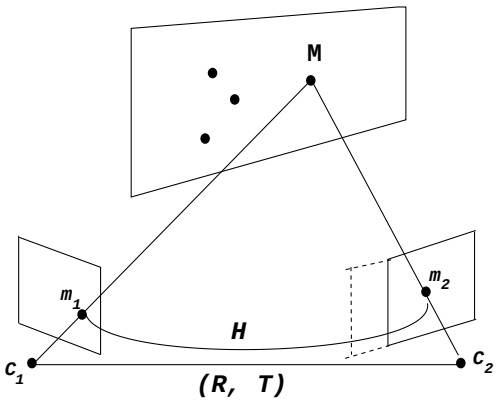


Figure 1. Planar projective transformation

The third type uses a set of images. Since it does not require calibration patterns, the images can be natural scenes. Stein has proposed a method which uses only images of the scene[12]. This method, however, requires a computer-driven rotating table to provide the rotating angle between images. Stein has also proposed a method which does not need calibration patterns nor rotating tables [13]. This method, however, cannot be applied directly to image mosaicing because its purpose is for 3D reconstruction. Another problem of this method is that the computational cost is high because it is based on a non-linear minimization framework.

Sawhney and Kumar have proposed a method for image mosaicing[15]. This method is an extension of direct methods for image mosaicing, and incorporates the radial distortion parameter into the homography computation between images. The problem of this method is that its computational cost is high. Since the method is based on iterative non-linear minimization, it requires iterative image warpings to estimate both homography and distortion parameters.

We propose a feature-based method for image mosaicing that has the following advantages. First, the method does not require any calibration pattern. Second, it is faster than previous methods.

2 Image mosaicing using homography

Given two images taken from the same viewpoint, or images of a planar scene taken from different viewpoints, the relationship between the images can be described by a planar projective transformation called homography [6]. This is the most accurate geometric transformation between images, and considers perspective effects.

Figure 1 illustrates the principal of the planar projective transformation for image mosaicing. When we observe a point M on a planar surface from two different viewpoints C_1 and C_2, we can transform the image coordinates $\mathbf{m_1} = (x_1, y_1, 1)^t$ to $\mathbf{m_2} = (x_2, y_2, 1)^t$ using the following 3×3 planar projective transformation matrix $\mathbf{H}$.

$$k\mathbf{m_2} = \mathbf{H}\mathbf{m_1} \tag{1}$$

where k is a non-zero arbitrary scale-factor and image coordinates $\mathbf{m_1}$ and $\mathbf{m_2}$ are represented by homogeneous coordinates. This relationship can be rewritten using the following equations.

$$\begin{cases} x_2 = \frac{h_0 x_1 + h_1 y_1 + h_2}{h_6 x_1 + h_7 y_1 + 1} \\ y_2 = \frac{h_3 x_1 + h_4 y_1 + h_5}{h_6 x_1 + h_7 y_1 + 1} \end{cases} \tag{2}$$

When a point on the planar surface is invisible from C_2 but visible from C_1, we can generate the corresponding point on image I_2 by this transformation.

To compute a homography from images, a lot of methods have already been proposed for direct methods, [16, 9, 17, 15], feature-based methods, [21, 3], and a phase-based method, [5]. Most of the methods except [3] are slow in computing homography. We propose a feature-based method whose computational cost is lower than previous methods, not only for computing homographies, but also for computing radial distortion parameters.

3 Radial lens distortion

3.1 Error function for distortion parameter

Let (u_t, v_t) be the true (distortion-free) pixel image coordinates, and (u_a, v_a) the corresponding actual (observed) image coordinates. We then have the following equations:

$$\begin{aligned} u_a &= u_t + (u_t - u_0)kr^2 \\ v_a &= v_t + (v_t - v_0)kr^2 \\ r^2 &= (u_t - u_0)^2 + (v_t - v_0)^t \end{aligned} \tag{3}$$

where k is the coefficient of the radial distortion and (u_0, v_0) are the center coordinates of the distortion [2].

When we have points $\mathbf{p}_i$ in the first image and their corresponding points $\mathbf{p}'_i$ in the second image related by the homography H, we define the sum of reprojection errors E as follows:

$$E = \sum_i (\mathbf{p}_i - H\mathbf{p}'_i)^2. \tag{4}$$

If we have a perfect lens without any lens distortion and with accurate feature correspondences, the error

should be zero. In fact, the error is not zero due to the lens distortion. We redefine the error function considering lens distortion. If we assume that the lens distortion k does not change between images with constant center coordinates (u_0, v_0) , we can rewrite the error E as follows:

$$E(k) = \sum_i (\mathbf{p}_i(k) - H\mathbf{p}'_i(k))^2. \quad (5)$$

After estimating the homography matrix H with the actual image coordinates, we can estimate the corresponding coefficient k by minimizing the error E.

We use the Newton-Raphson method to solve the error E in terms of k. When the change δk of the coefficient k is small, we have the following equation by Taylor expansion.

$$E(k + \delta k) = E(k) + \delta k E'(k) \quad (6)$$

where E' is the first derivative of the error E. Since we need to find δk to minimize the error E, we obtain the following equation.

$$E(k + \delta k) = 0 \quad (7)$$

By substituting the equation (7) into (6), we can estimate δk as follows:

$$\delta k = -\frac{E(k)}{E'(k)}. \quad (8)$$

To improve the accuracy, we can estimate k iteratively as follows:

$$k_{i+1} = k_i - \frac{E(k_i)}{E'(k_i)}. \quad (9)$$

We re-estimate the homography matrix from the corrected image coordinates by using the estimated distortion parameter k during this iteration. While the initial estimate of the homography might include the distortion effect, this re-estimation can gradually separate the initial homography into a homography for real image coordinates and the distortion parameter. Consult the appendix to investigate how to compute the derivative of error E.

3.2 Feature correspondence

We establish feature correspondence automatically by using optical flow estimation. We show the algorithm as follows:

1. Rough alignment
   We use correlation with low-resolution images.
2. Feature selection
   We select prominent features such as corners.
3. Coarse-to-fine optical flow estimation
   We use the Lucas-Kanade method for resolution pyramids
4. Feature correspondence
   We establish feature correspondence by using the flow estimate in subpixels.

Readers can find the details in [4].

We compute the homography between the two images from four or more feature correspondences with a least-squares method. To exclude false correspondences, we use a robust technique, the M-estimator. We can determine whether a feature point $\mathbf{p}$ has false correspondence or not, by measuring the re-projected error E_r as follows:

$$E_r = (\mathbf{p} - H(\mathbf{p}'))^2. \quad (10)$$

If the error E_r is above the pre-determined threshold, the corresponding point may have false correspondence. After excluding this point from the set of feature correspondences, we then compute the homography H again with the remaining features. We repeat this process until the errors of all points are smaller than the threshold value. This simple implementation works well, although many variations have been proposed as M-estimators.

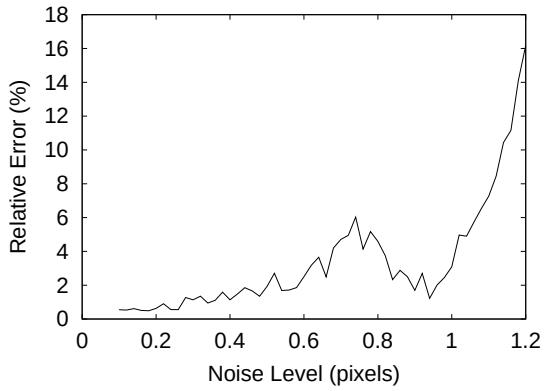
To increase the accuracy, we use a multi-resolution pyramid of images. After computing a homography at some level, we warp the image at the next finer level by using the estimated homography. At each level, we establish feature correspondence to estimate the homography. After estimating the homography at the finest level, we use the homography and the feature correspondences to compute the radial distortion coefficient k as well.

4 Experiments

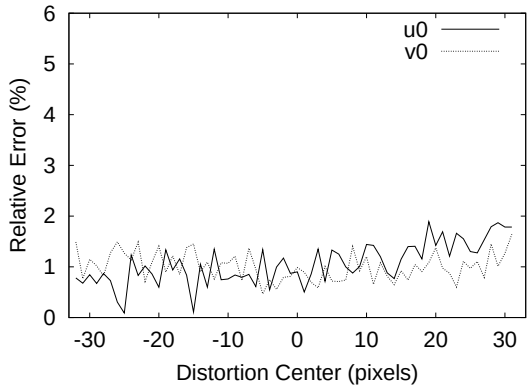
We conducted experiments on both synthetic images and real images. We show the performance analysis with regard to noise by computer simulation. On real images, we show the effectiveness of our method in accuracy and efficiency.

4.1 Computer simulations

We conducted two experiments on synthetic data. The image size of the simulated camera is 640 by 480 pixels. The focal length is 640 in pixels. We placed



Errors by varying noise levels



Errors by varying the center coordinates

Figure 2. Error analysis in simulation.

points in 3D at equal distance from the optical center. The points were aligned 50 pixels apart on the first image plane. Setting the distortion parameter to -3.0e-7 as the ground true value, we projected the 3D points on the first image. The distortion value was determined by experiments with an actual zooming lens. Then we rotated the camera horizontally around the optical center by 30 degrees for the second image. This gives us the true homography between the images. Half of the image plane was overlapped with the first image. We obtained 37 pairs of corresponding points in the overlapping region between the images. These correspondences contain radial distortion.

In the first experiment, we evaluated the performance with regard to the noise level in feature correspondence. We added gaussian noise with 0 mean and σ standard deviation to the point coordinates on the second image. We varied the noise level, σ, from 0.1 pixel to 1.2 pixels. For each noise level, we performed 100 independent trials, and the results were shown in an average. We compared the estimated lens distortion coefficient with the ground truth. In the experiments, the homography was also estimated from the feature correspondences. Figure 2 top shows the errors at different noise levels. As we can see from the figure, the errors were less than 5 % until the 0.5 noise level.

In the second experiment, we evaluated the performance with regard to the deviation of the center coordinates of the radial lens distortion. Figure 2 bottom shows the errors. Fixing the noise level at 0.3 pixel, we varied the center coordinates from -32.0 to 32.0 pixels in both directions, horizontal $u0$ and vertical $v0$. Although our method assumes that the center of the distortion equals the center of the image, the errors to the offset of the distortion center were less than 2 %, which was smaller than we expected.

4.2 Real data

We conducted two experiments on real images with two types of lenses, zoom and wide-angle. First, we show the result with a zooming lens. Two images were taken by a hand-held digital still camera (Nikon CoolPix 950) that is equipped with a zooming lens. The overlap of the images was around 50 % in vertical direction. We obtained the lens distortion coefficient as -3.19e-7 from 23 pairs of feature correspondences. The error E has been reduced from 13.3 to 3.1, which is 23.3 %.

Figure 3 shows the effectiveness of correcting lens distortion for image mosaicing. Figure 3 left shows the mosaic without lens distortion correction. Figure 3 right shows the mosaic corrected by the estimated distortion coefficient with the proposed method. While the edges of the book are curved in the left image, those in the right image are straight.

In the second experiment, we show the result with a wide-angle lens. Two images were taken with a wide-angle conversion lens (Olympus WCON-8), that is attached to a digital camera (Olympus C-3030ZOOM). The focal length is 25.6mm in 35mm-film equivalent. The overlap between the images was around 50 % in horizontal direction. We established 70 pairs of feature correspondences. The estimated lens distortion coefficient was -4.95e-7. The error E has been reduced from 160.6 to 116.8, which is 72.7%. Figure 4 right shows the corrected image of the second image. While the pillar is curved in the original image (left), it is straight in the corrected image (right). Figure 5 shows another example of correcting radial distortion by using the estimated value. The figure right shows one of the corrected input images. Although the edges of the side wall behind the sink are curved in the original (left), they are straight in the corrected image (right).

Original Corrected

Figure 4. Corrected input image (Wide-angle: Olympus C-3030 + FCON-08)

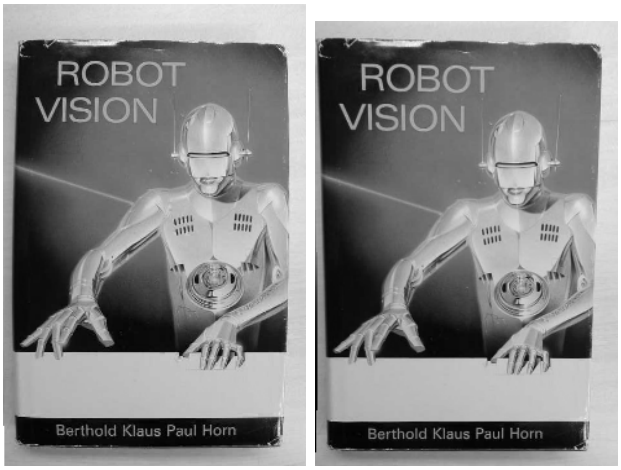


Figure 3. The left shows the mosaic without lens distortion correction. The right shows the mosaic with lens distortion correction.

Table 1. Error Comparisons

Lens	Method	k	Error
Zoom	Proposed	-3.19e-7	3.10
	Zhang	-3.52e-7	3.22
Wide-angle	Proposed	-4.95-7	116.8
	Zhang	-6.94e-7	212.2

4.3 Comparison

We compare the accuracy and efficiency of our method with previous methods. First, we compare the accuracy of our method with a previous method developed by Zhang [20]. This method is supposed to be more accurate than our method, because it uses a planar calibration pattern and more images (five) than our method (two). Tables 1 shows comparisons with two different cameras. We used natural scenes to estimate the distortion parameter for our method. We used the images shown in Figure 3 for a zooming camera (CoolPix 950) and Figure 4 for a wide-angle camera (C-3030 + FCON-08).

For Zhang's method, we used a calibration pattern which has 108 circles printed on a sheet of paper by a laser printer. The circles were aligned in a 12 by 9 formation with displacements of 20 mm. The center coordinates of each circle was computed after the circle pixels were extracted by binarization followed by a labeling technique. We used five images taken from different viewpoints to estimate the internal camera parameters, including the radial distortion parameter. Our implementation does not take into account the second term of radial distortion coefficients and assumes that the skew parameter is zero.

Tables 1 shows the sum of reprojected errors in Equation (5). From the tables, we can see that the errors of our method are smaller than those of Zhang's method.

Second, we compare the computational efficiency of our method with a previous method, developed in [15]. The previous method is slow to compute because it is based on a non-linear minimization framework. Table 2 shows the processing time for images taken with the two types of cameras. We measured the processing

Original Corrected

Figure 5. Corrected input image (Wide-angle)

Table 2. Processing time (seconds)

Method	Zoom	Wide-angle
Proposed	12.1	11.0
Non-linear	41.6	45.6

time on an SGI Indigo 2 (195MHz). The results show that our method was about four times faster than the previous method.

In the experiment, we measured the processing time with the following conditions. The results of rough alignment were provided to both methods. The resolution pyramid of four levels was prepared in both methods for hierarchical estimation. In the previous method, the maximum number of iterations at each level was set to 10. We measured the processing time for computing the radial distortion parameter and the homography, not including the time for rough alignment and the final mosaic construction. In our method, the additional time of computing distortion coefficient k to homography computation was less than one second.

5 Discussion

Our method is about four times faster than the previous method proposed in [15]. The previous method is an extension of direct methods that use non-linear minimization, not requiring calibration patterns. The idea of our method is similar to that of the previous method, except that the definition of error function is different. This difference makes our method faster than the previous method because our method does not require iterative image warpings.

The major difference in computational cost is the number of iterative image warpings. Our method requires only three image warpings for hierarchical estimation. On the other hand, the previous method might require forty iterations at most. While flow estimation in our method by using the Lucas-Kanade method requires iterative computation, it requires no image warpings. Since it is based on 2D translation, the computational cost for computing image coordinates for each pixel is two additions. On the other hand, the previous method requires image warping with more parameters at each iteration. We warp images by using the currently estimated homography and distortion coefficient pixel-by-pixel. The computational cost is

higher than 2D translation, because the computation of warping by homography consists of 6 multiplications, 6 additions and 2 divisions, to compute the image coordinates for each pixel. In addition, the previous method needs to compute image derivatives for each warped image. This process increases the computational cost. Furthermore, the cost for computing the partial derivatives of 9 parameters, homography and distortion coefficient, cannot be negligible. Our method, however, requires feature extraction. The computational cost of this part is less than that of the image warpings in the previous method.

The accuracy of the method is comparable to previous methods using special calibration patterns. Using only images of natural scenes taken with a hand-held camera, we were able to estimate a reasonable distortion parameter. We might be able to recover other parameters such as image center coordinates. By simulation, however, we confirmed that the slight offset of the image center coordinates is negligible. To compute the second term of the distortion parameters is a subject of future research.

We can use more than two images in order to increase the accuracy of estimating the lens distortion parameter. To do this, we can define the error function as follows.

$$E(k) = \sum_j \sum_i (\mathbf{p}_i^j(k) - H(\mathbf{p}_i^{j+1}(k)))^2 \qquad (11)$$

where $\mathbf{p}_i^j$ represents the ith image coordinates in the jth image. An experiment to analyze this performance is another subject of future research.

6 Conclusion

In this paper, we have illustrated a method for automatically correcting radial lens-distortion for image mosaicing. The method has two advantages over existing techniques. First, it does not require any special calibration patterns. We showed that the accuracy of the proposed method is comparable to the previous method using a calibration pattern. Second, it is faster than the previous method. Experiments showed that it is about four times faster than the previous method.

In future work, we plan to analyze the effectiveness of considering a second distortion parameter k_2 for image mosaicing.

Acknowledgement

We thank Takashi Iida for his code to extract circles for Zhang's method. We also thank Dr. Masashi Yasuda, department director, and Mr. Matatsugu Honjo, center director, for giving us an opportunity to proceed this research.

References

[1] J. R. Bergen et al., "Hierarchical model-based motion estimation," Proc. Second European Conf. Computer Vision, pp. 237-252, 1992.

[2] D. C. Brown, "Close-range camera calibration," Photogrammetric Engineering. Vol. 37, No.8, pp. 855-866, 1971.

[3] D. Capel and A. Zisserman, "Automated mosaicing with super-resolution zoom," Proc. Conf. on Computer Vision and Pattern Recognition, pp. 885-891, 1998.

[4] N. Chiba, H. Kano, et al "Feature-based image mosaicing", Proc. IAPR Workshop on Machine Vision Applications, pp. 5-10, 1998.

[5] J. Davis, "Mosaics of scenes with moving objects," Proc. Conf. on Computer Vision and Pattern Recognition, pp. 354-360, 1998.

[6] O. Faugeras, "Three-dimensional computer vision: A geometric viewpoint," MIT Press, 1993.

[7] R. Kumar, P. Anandan and K. Hanna, "Direct recovery of shape from multiple views: a parallax based approach," Proc. International Conf. on Pattern Recognition, pp. 685 -688, 1994.

[8] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," In the Seventh International Joint Conference on Artificial Intelligence (IJCAI-81), pp. 674-679, 1981.

[9] S. Mann and R. W. Picard, "Virtual bellows: constructing high quality stills from video," Proc. International Conf. on Image Processing, pp. 363-367, 1994.

[10] M. Irani, P. Anandan and S. Hsu, "Mosaic based representations of video sequences and their applications," Proc. International Conf. on Computer Vision, pp. 605-611, 1995.

[11] R. Swaminathan and S. K. Nayar, "Non-metric calibration for wide-angle lenses and polycameras," Proc. Conf. on Computer Vision and Pattern Recognition, pp. 413-419, 1999.

[12] G. P. Stein, "Accurate internal camera calibration using rotation, with analysis of sources of error," Proc. International Conf. on Computer Vision, pp. 230-236, 1995.

[13] G. P. Stein, "Lens distortion calibration using point correspondences," Proc. Conf. on Computer Vision and Pattern Recognition, pp. 602-608, 1997.

[14] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off the shelf TV cameras and lenses," IEEE Journal of Robotics and Automation, RA-3, pp.323-344, August, 1987.

[15] H. S. Sawhney and R. Kumar, "True multi-image alignment and its application to mosaicing and lens distortion correction," IEEE PAMI Vo. 21, No. 3, pp. 235-243, 1999.

[16] R. Szeliski, "Image mosaicing for tele-reality applications, " IEEE Workshop Applications of Computer Vision, pp. 44-53, 1994.

[17] R. Szeliski and H. Y. Shum, "Creating full view panoramic image mosaics and environment maps," Proc. SIGGRAPH '97, pp. 251-258, 1997.

[18] S. Peleg and J. Herman, "Panoramic mosaics by manifold projection, " Proc. Conf. Computer Vision and Pattern Recognition, pp. 338-343, 1997.

[19] B. Rousso, S. Peleg et al., "Universal mosaicing using pipe projection," Proc. International Conf. on Computer Vision, pp. 945-952, 1998.

[20] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," Proc. International Conf. on Computer Vision, pp. 666-673, 1999.

[21] I. Zoghlami, O. Faugeras and R. Deriche, "Using geometric corners to build a 2D mosaic from a set of images," Proc. Conf. Computer Vision and Pattern Recognition, pp. 420-425, 1997.

Appendix

The derivative of error E

Here we show how to compute the derivative of the error E in Equation (5). To distinguish true points from the distorted points, we rewrite the error as follows:

$$E = \sum (\mathbf{p_t} - H\mathbf{p_t}')^2 \tag{12}$$

where $\mathbf{p_t}$ is a distortion free, true point and a function of the distortion parameter k.

The derivative of the error E can be expressed as:

$$\frac{\delta E}{\delta k} = \sum 2(\mathbf{p_t} - H\mathbf{p_t}') \left(\frac{\delta \mathbf{p_t}}{\delta k} - \frac{\delta H\mathbf{p_t}'}{\delta k} \right). \tag{13}$$

We show how to compute each term. For the first term, we need to obtain true points, because we have only distorted points as follows:

$$\mathbf{p_d} = \mathbf{p_t} + \mathbf{p_t} k {r_t}^2 \tag{14}$$

where $\mathbf{p_t}$ is a true point, $\mathbf{p_d}$ is its distorted point by the radial distortion k, and ${r_t}^2$ is the squared distance from the image center for the true point. We obtain the true points from the distorted points by using the current value of the distortion parameter k. If we assume that the squared distance of a true point is nearly equals to that of the distorted point, we can obtain the true point as follows:

$$\mathbf{p_t} = \frac{\mathbf{p_d}}{1 + k{r_d}^2} \tag{15}$$

where ${r_d}^2$ is the squared distance for the distorted point. By using this equation, we can obtain the derivative of points with respect to the distortion parameter k in the second term as follows:

$$\frac{\delta \mathbf{p_t}}{\delta k} = -\mathbf{p_d} C \tag{16}$$

where

$$C = \frac{{r_d}^2}{(1 + k{r_d}^2)^2}.$$

The derivative of the transformed points by Equation (2) in true coordinate system can be expressed as follows:

$$\frac{\delta H\mathbf{p_t}'}{\delta k} = \begin{bmatrix} \frac{\delta}{\delta k}\left(\frac{A}{D}\right) \\ \frac{\delta}{\delta k}\left(\frac{B}{D}\right) \end{bmatrix} \tag{17}$$

where

$$\begin{aligned} A &= h_0 {x_t}' + h_1 {y_t}' + h_2 \\ B &= h_3 {x_t}' + h_4 {y_t}' + h_5 \\ D &= h_6 {x_t}' + h_7 {y_t}' + 1 \end{aligned}$$

By using Equation (15), we can rewrite the above equation with the distorted points as follows:

$$\begin{aligned} A &= h_0 {x_d}' G + h_1 {y_d}' G + h_2 \\ B &= h_3 {x_d}' G + h_4 {y_d}' G + h_5 \\ D &= h_6 {x_d}' G + h_7 {y_d}' G + 1 \end{aligned}$$

where

$$G = \frac{1}{1 + k{r_d}^2}.$$

The derivatives in Equation (17) can be obtained as follows:

$$\begin{aligned} \frac{\delta}{\delta k}\left(\frac{A}{D}\right) &= \frac{\delta A}{\delta k} D - \frac{A}{D^2}\frac{\delta D}{\delta k} \\ \frac{\delta}{\delta k}\left(\frac{B}{D}\right) &= \frac{\delta B}{\delta k} D - \frac{B}{D^2}\frac{\delta D}{\delta k} \end{aligned}$$

where

$$\begin{aligned} \frac{\delta A}{\delta k} &= -h_0 {x_d}' C - h_1 {y_d}' C \\ \frac{\delta B}{\delta k} &= -h_2 {x_d}' C - h_3 {y_d}' C \\ \frac{\delta D}{\delta k} &= -h_6 {x_d}' C - h_7 {y_d}' C \\ C &= \frac{{r_d}^2}{(1 + k{r_d}^2)^2}. \end{aligned}$$

Now we have all of the elements to solve the derivative of the error E in Equation (13).